

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**EV355228983**

**Multimedia Timeline Modification in Networked  
Client/Server Systems**

Inventor(s):  
Anoop Gupta  
Nosakhare D. Omoigui

ATTORNEY'S DOCKET NO. MS1-272USC2



1 **RELATED APPLICATIONS**

2 This is a continuation of U.S. Patent Application Serial No. 09/153,664,  
3 filed September 15, 1998, entitled "Multimedia Timeline Modification in  
4 Networked Client/Server Systems", to Anoop Gupta and Nosakhare D. Omigui.

5  
6 **TECHNICAL FIELD**

7 This invention relates to networked client/server systems and to methods of  
8 streaming and rendering multimedia content in such systems.

9  
10 **BACKGROUND OF THE INVENTION**

11 Multimedia streaming—the continuous delivery of synchronized media  
12 data like video, audio, text, and animation—is a critical link in the digital  
13 multimedia revolution. Today, streaming media is primarily about video and  
14 audio, but a richer, broader digital media era is emerging with a profound and  
15 growing impact on the Internet and digital broadcasting.

16 Synchronized media means multiple media objects that share a common  
17 timeline. Video and audio are examples of synchronized media—each is a  
18 separate data stream with its own data structure, but the two data streams are  
19 played back in synchronization with each other. Virtually any media type can  
20 have a timeline. For example, an image object can change like an animated .gif  
21 file: text can change and move, and animation and digital effects happen over  
22 time. This concept of synchronizing multiple media types is gaining greater  
23 meaning and currency with the emergence of more sophisticated media  
24  
25



1 composition frameworks implied by MPEG-4, Dynamic HTML, and other media  
2 playback environments.

3 The term “streaming” is used to indicate that the data representing the  
4 various media types is provided over a network to a client computer on a real-  
5 time, as-needed basis, rather than being pre-delivered in its entirety before  
6 playback. Thus, the client computer renders streaming data as it is received from a  
7 network server, rather than waiting for an entire “file” to be delivered.

8 The widespread availability of streaming multimedia enables a variety of  
9 informational content that was not previously available over the Internet or other  
10 computer networks. Live content is one significant example of such content.  
11 Using streaming multimedia, audio, video, or audio/visual coverage of noteworthy  
12 events can be broadcast over the Internet as the events unfold. Similarly,  
13 television and radio stations can transmit their live content over the Internet.

14 In comparison to text-based or paper-based presentations, streaming  
15 multimedia presentations are very effective in certain situations. Audio/visual  
16 presentations, for example, are able to capture and convey many subtle factors that  
17 are not perceivable from paper-based documents. Even when the content is a  
18 spoken presentation, an audio/visual recording captures gestures, facial  
19 expressions, and various speech nuances that cannot be discerned from text or  
20 even from still photographs.

21 Although streaming multimedia content compares favorably with textual  
22 content in most regards, one disadvantage is that it requires significant time for  
23 viewing. It cannot be “skimmed” like textual content. Thus, information  
24  
25



1 consumers are forced to choose between the efficiency of the written word and the  
2 richness of the multimedia experience.

3 The invention described below addresses this disadvantage of prior art  
4 streaming multimedia content, allowing more efficient multimedia perusal of  
5 streaming multimedia presentations than has previously been possible.

### 6 7 **SUMMARY OF THE INVENTION**

8 The invention utilizes time-scale modification so that a user can vary the  
9 speed of streaming content without destroying its intelligibility. In accordance  
10 with the invention, a user selects multimedia content from a menu presented at a  
11 network client computer. In addition, the user selects a speed factor, indicating the  
12 speed at which the multimedia should be rendered relative to its default speed.

13 In response to these selections by the user, the network client contacts a  
14 network server and initiates a composite media stream from the server to the  
15 client. As it receives the media stream, the client processes the individual media  
16 streams of the composite stream, altering their timelines in accordance with the  
17 user's speed designation. Time-scale modification techniques are used to alter  
18 audio streams to maintain their original pitch.

19 In alternative embodiments, timelines are altered at the server, before  
20 streaming to the client. To accomplish this, the server either alters the timelines  
21 dynamically, as specific content is requested, or selects from a number of pre-  
22 stored media streams having pre-altered timelines.

23 The invention includes methods of adapting to limited bandwidth situations  
24 by composing or selecting composite streams having differing degrees of quality,  
25



1 and/or by composing or selecting streams with timelines that are altered to closely  
2 correspond with whatever speed factor has been chosen. In one embodiment of  
3 the invention, certain media streams, such as audio streams, take precedence over  
4 other streams such as video streams. In this embodiment of the invention, the  
5 audio stream is sent with an unaltered timeline, at a rate sufficient to satisfy the  
6 consumption requirements of the client, given the current speed factor. The video  
7 is then degraded in quality to reduce its bandwidth, so that it can be streamed in  
8 whatever bandwidth is not require by the audio.

#### 9 10 **BRIEF DESCRIPTION OF THE DRAWINGS**

11 Fig. 1 is a block diagram of a networked client/server system in accordance  
12 with the invention.

13 Fig. 2 is a block diagram of a networked computer that can be used to  
14 implement either a server or a client in accordance with the invention.

15 Fig. 3 is a block diagram illustrating communications and rendering of a  
16 composite media stream in accordance with the invention.

17 Fig. 4 is a flowchart illustrating methodological aspects of the invention.

18 Fig. 5 is a diagrammatic illustration of a graphical user interface window  
19 having a time-scale-modification tool for changing a playback speed of streaming  
20 multimedia content according to one implementation.

21 Fig. 6 is a diagrammatic illustration of a graphical user interface window  
22 having a time-scale-modification tool according to a second implementation.

23 Fig. 7 is a diagrammatic illustration of a graphical user interface window  
24 having a time-scale-modification tool according to a third implementation.  
25



1 Fig. 8 is a block diagram illustrating one embodiment in which a plurality  
2 of timeline-altered media streams are stored at and provided from a server.

3 Fig. 9 is a block diagram illustrating another embodiment in which a  
4 plurality of timeline-altered media streams are stored at and provided from a  
5 server.

6 Fig. 10 is a block diagram illustrating yet another embodiment in which a  
7 plurality of timeline-altered media streams are stored at and provided from a  
8 server.

9 Fig. 11 is a block diagram illustrating yet another embodiment in which a  
10 plurality of timeline-altered media streams are stored at and provided from a  
11 server.

## 12 13 **DETAILED DESCRIPTION**

### 14 **General Network Structure**

15 Fig. 1 shows a client/server network system and environment in accordance  
16 with the invention. Generally, the system includes a network server computer 10  
17 and a plurality of network client computers 11. The computers communicate with  
18 each other over a data communications network. The communications network in  
19 Fig. 1 comprises a public network 12 such as the Internet. The data  
20 communications network might also include local-area networks and private wide-  
21 area networks.

22 Server computer 10 has access to streaming media content in the form of  
23 different composite media streams. Some composite media streams might be  
24 stored as files in a database or other file storage system 13. Other composite  
25



1 media streams might be supplied to the server on a “live” basis from other data  
2 source components through dedicated communications channels or through the  
3 Internet itself.

#### 4 5 **Streaming Media**

6 In this discussion, the term “composite media stream” describes  
7 synchronized streaming data that represents a segment of multimedia content. The  
8 composite media stream has a timeline that establishes the speed at which the  
9 content is rendered. The composite media stream can be rendered to produce a  
10 plurality of different types of user-perceivable media, including synchronized  
11 audio or sound, video graphics or motion pictures, animation, textual content,  
12 command script sequences, or other media types that convey time-varying  
13 information or content in a way that can be sensed and perceived by a human. A  
14 composite media stream comprises a plurality of individual media streams  
15 representing the multimedia content. Each of the individual media streams  
16 corresponds to and represents a different media type and each of the media streams  
17 can be rendered by a network client to produce a user-perceivable presentation  
18 using a particular presentation medium. The individual media streams have their  
19 own timelines, which are synchronized with each other so that the media streams  
20 can be rendered simultaneously for a coordinated multimedia presentation.

21 There are various standards for streaming media content and composite  
22 media streams. The “Advanced Streaming Format” (ASF) is an example of such a  
23 standard, including both accepted versions of the standard and proposed standards  
24 for future adoption. ASF specifies the way in which multimedia content is stored,  
25



1 streamed, and presented by the tools, servers, and clients of various multimedia  
2 vendors. ASF provides benefits such as local and network playback, extensible  
3 media types, component download, scalable media types, prioritization of streams,  
4 multiple language support, environment independence, rich inter-stream  
5 relationships, and expandability. Further details about ASF are available from  
6 Microsoft Corporation of Redmond, Washington.

7       Regardless of the streaming format used, an individual data stream contains  
8 a sequence of digital data units that are rendered individually, in sequence, to  
9 produce an image, sound, or some other stimuli that is perceived by a human to be  
10 continuously varying. For example, an audio data stream comprises a sequence of  
11 sample values that are converted to a pitch and volume to produce continuously  
12 varying sound. A video data stream comprises a sequence of digitally-specified  
13 graphics frames that are rendered in sequence to produce a moving picture.

14       Typically, the individual data streams of a composite media stream are  
15 interleaved in a single sequence of data packets. Various types of data  
16 compression might be used within a particular data format to reduce  
17 communications bandwidth requirements.

18       The sequential data units (such as audio sample values or video frames) are  
19 associated with both delivery times and presentation times, relative to an arbitrary  
20 start time. The delivery time of a data unit indicates when the data unit should be  
21 delivered to a rendering client. The presentation time indicates when the value  
22 should be actually rendered. Normally, the delivery time of a data unit precedes  
23 the presentation time.



1       The presentation times determine the actual speed of playback. For data  
2 streams representing actual events or performances, the presentation times  
3 correspond to the relative times at which the data samples were actually recorded.  
4 The presentation times of the various different individual data streams are  
5 consistent with each other so that the streams remain coordinated and  
6 synchronized during playback.

### 7 8 **Exemplary Computer Environment**

9       In the discussion below, the invention will be described in the general  
10 context of computer-executable instructions, such as program modules, being  
11 executed by one or more conventional personal computers. Generally, program  
12 modules include routines, programs, objects, components, data structures, etc. that  
13 perform particular tasks or implement particular abstract data types. Moreover,  
14 those skilled in the art will appreciate that the invention may be practiced with  
15 other computer system configurations, including hand-held devices,  
16 multiprocessor systems, microprocessor-based or programmable consumer  
17 electronics, network PCs, minicomputers, mainframe computers, and the like. In a  
18 distributed computer environment, program modules may be located in both local  
19 and remote memory storage devices.

20       Fig. 2 shows a general example of a computer 20 that can be used as a  
21 network node or host in accordance with the invention. Computer 20 is shown as  
22 an example of a computer that can perform the functions of either server computer  
23 10 or a client computer 11.



1 Computer 20 includes one or more processors or processing units 21, a  
2 system memory 22, and a system bus 23 that couples various system components  
3 including the system memory to processors 21.

4 The system bus 23 may be any of several types of bus structures including a  
5 memory bus or memory controller, a peripheral bus, and a local bus using any of a  
6 variety of bus architectures. The system memory includes read only memory  
7 (ROM) 24 and random access memory (RAM) 25. A basic input/output system 26  
8 (BIOS), containing the basic routines that help to transfer information between  
9 elements within server computer 20, such as during start-up, is stored in ROM 24.  
10 Computer 20 further includes a hard disk drive 27 for reading from and writing to  
11 a hard disk, not shown, a magnetic disk drive 28 for reading from or writing to a  
12 removable magnetic disk 29, and an optical disk drive 30 for reading from or  
13 writing to a removable optical disk 31 such as a CD ROM or other optical media.  
14 The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are  
15 connected to the system bus 23 by an SCSI interface 32 or some other appropriate  
16 interface. The drives and their associated computer-readable media provide  
17 nonvolatile storage of computer readable instructions, data structures, program  
18 modules and other data for computer 20. Although the exemplary environment  
19 described herein employs a hard disk, a removable magnetic disk 29 and a  
20 removable optical disk 31, it should be appreciated by those skilled in the art that  
21 other types of computer readable media which can store data that is accessible by a  
22 computer, such as magnetic cassettes, flash memory cards, digital video disks,  
23 random access memories (RAMs) read only memories (ROM), and the like, may  
24 also be used in the exemplary operating environment.  
25



1 A number of program modules may be stored on the hard disk, magnetic  
2 disk 29 optical disk 31, ROM 24, or RAM 25, including an operating system 35,  
3 one or more application programs 36, other program modules 37, and program  
4 data 38. A user may enter commands and information into computer 20 through  
5 input devices such as keyboard 40 and pointing device 42. Other input devices  
6 (not shown) may include a microphone, joystick, game pad, satellite dish, scanner,  
7 or the like. These and other input devices are connected to the processing unit 21  
8 through an interface 46 that is coupled to the system bus. A monitor 47 or other  
9 type of display device is also connected to the system bus 23 via an interface, such  
10 as a video adapter 48. In addition to the monitor, personal computers typically  
11 include other peripheral output devices (not shown) such as speakers and printers.

12 Computer 20 operates in a networked environment using logical  
13 connections to one or more remote computers, such as a remote computer 49. The  
14 remote computer 49 may be another personal computer, a server, a router, a  
15 network PC, a peer device or other common network node, and typically includes  
16 many or all of the elements described above relative to computer 20, although only  
17 a memory storage device 50 has been illustrated in Fig. 2. The logical connections  
18 depicted in Fig. 2 include a local area network (LAN) 51 and a wide area network  
19 (WAN) 52. Such networking environments are commonplace in offices,  
20 enterprise-wide computer networks, intranets, and the Internet. In the described  
21 embodiment of the invention, remote computer 49 executes an Internet Web  
22 browser program such as the "Internet Explorer" Web browser manufactured and  
23 distributed by Microsoft Corporation of Redmond, Washington.



1 When used in a LAN networking environment, computer 20 is connected to  
2 the local network 51 through a network interface or adapter 53. When used in a  
3 WAN networking environment, computer 20 typically includes a modem 54 or  
4 other means for establishing communications over the wide area network 52, such  
5 as the Internet. The modem 54, which may be internal or external, is connected to  
6 the system bus 23 via a serial port interface 33. In a networked environment,  
7 program modules depicted relative to the personal computer 20, or portions  
8 thereof, may be stored in the remote memory storage device. It will be  
9 appreciated that the network connections shown are exemplary and other means of  
10 establishing a communications link between the computers may be used.

11 Generally, the data processors of computer 20 are programmed by means of  
12 instructions stored at different times in the various computer-readable storage  
13 media of the computer. Programs and operating systems are typically distributed,  
14 for example, on floppy disks or CD-ROMs. From there, they are installed or  
15 loaded into the secondary memory of a computer. At execution, they are loaded at  
16 least partially into the computer's primary electronic memory. The invention  
17 described herein includes these and other various types of computer-readable  
18 storage media when such media contain instructions or programs for implementing  
19 the steps described below in conjunction with a microprocessor or other data  
20 processor. The invention also includes the computer itself when programmed  
21 according to the methods and techniques described below. Furthermore, certain  
22 sub-components of the computer may be programmed to perform the functions  
23 and steps described below. The invention includes such sub-components when  
24 they are programmed as described. In addition, the invention described herein  
25



1 includes data structures, described below, as embodied on various types of  
2 memory media.

3 For purposes of illustration, programs and other executable program  
4 components such as the operating system are illustrated herein as discrete blocks,  
5 although it is recognized that such programs and components reside at various  
6 times in different storage components of the computer, and are executed by the  
7 data processor(s) of the computer.

### 8 9 **Client-Based Multimedia Time-Scale Modification**

10 As shown in Fig. 1, a network system in accordance with the invention  
11 includes a network server 10 from which a plurality of composite media streams  
12 are available. In some cases, the composite media streams are actually stored by  
13 server 10. In other cases, server 10 obtains the composite media streams from  
14 other network sources or devices.

15 The system also includes network clients 11. Generally, the network clients  
16 are responsive to user input to select or request identified composite media  
17 streams. In response to a request for a composite media stream, server 10 streams  
18 the requested composite media stream to the network client in accordance with  
19 some known format such as ASF. The client renders the data streams to produce  
20 the multimedia content.

21 In accordance with the invention, a network client also accepts a speed  
22 designation from a human user. The speed designation is preferably a speed factor  
23 relative to the original or default playback speed of the selected multimedia  
24 stream. For example, a speed factor of 1.2 indicates that the composite media  
25



1 stream is to be rendered at 1.2 times its original or default speed, thereby  
2 achieving time compression. A speed factor of 0.8 indicates that the composite  
3 media stream is to be rendered at 0.8 times its original or default speed, thereby  
4 achieving time expansion.

5 In response to the speed designation from the user, the system modifies the  
6 timelines of the individual media streams of the composite media stream, while  
7 keeping the timelines synchronized with each other and while maintaining the  
8 original pitch of any audio produced from audio streams. In one embodiment of  
9 the invention, such timeline modification is performed by the network client. In  
10 other embodiments of the invention, the timeline modification can be performed at  
11 the network server, before the media streams are streamed to the network client.

12 Timeline modification changes the timeline of the received data streams in  
13 accordance with the user speed designation to achieve either time compression or  
14 time expansion. With some types of media, such as video streams, this involves  
15 either omitting selected frames or modifying the presentation times of the  
16 individual data units or video frames. In other cases, such as with audio streams,  
17 the time-modification is more difficult—simply changing the presentation times  
18 would alter the pitch of the original audio and make it unintelligible. Accordingly,  
19 some type of audio processing technique is used to time-compress or time-expand  
20 audio streams, while maintaining the original pitch of the audio—thereby  
21 maintaining the intelligibility of the audio.

22 There are various known methods of audio time modification, commonly  
23 referred to as “time-scale-modification,” most of which concentrate on removing  
24 redundant information from the speech signal. In a method referred to as  
25



1 sampling, short segments are dropped from the speech signal at regular intervals.  
2 Cross fading or smoothing between adjacent segments improves the resulting  
3 sound quality.

4 Another method, referred to as synchronized overlap add method (SOLA or  
5 OLA), consists of shifting the beginning of a new speech segment over the end of  
6 the preceding segment to find the point of highest cross-correlation (i.e., maximum  
7 similarity). The overlapping frames are averaged, or smoothed together, as in the  
8 sampling method.

9 Sampling with dichotic presentation is a variant of the sampling method  
10 that takes advantage of the auditory system's ability to integrate information from  
11 both ears. It improves on the sampling method by playing the standard sampled  
12 signal to one ear and the "discarded" material to the other ear. Intelligibility and  
13 compression increase under this dichotic presentation condition when compared  
14 with standard presentation techniques.

15 The methods mentioned above are considered "linear" because all portions  
16 of the speech signal are compressed or expanded uniformly. Other methods are  
17 considered non-linear because they non-uniformly remove portions of the time  
18 signal. One example of a non-linear time-compression method is referred to as  
19 pause removal. When using this method, a speed processing algorithm attempts to  
20 identify and remove any pauses in a recording. The embodiments described herein  
21 implement linear time-scale modification, although non-linear time-scale  
22 modification techniques can also be utilized.

23 More information regarding audio time modification is given in an article  
24 that appeared in the March, 1997, issue of "ACM Transactions on Computer-  
25



1 Human Interaction” (Volume 4, Number 1, pages 3-38) (1997). For purposes of  
2 this disclosure, it can be assumed that audio time modification involves some  
3 combination of changing individual data stream samples, dropping certain  
4 samples, and adjusting presentation times of any samples that are actually  
5 rendered.

6 Fig. 3 illustrates an embodiment of the invention in which timeline  
7 modification is performed by network client 11. Network server 10 streams a  
8 composite media stream 101 to network client 11 (although not shown in Fig. 3,  
9 other communications also take place bi-directionally between server 10 and client  
10 11, such as control-oriented communications). In addition to communications of  
11 the media stream 101, there are various control communications between the  
12 server. The composite media stream has a plurality of individual media streams as  
13 described above. For purposes of discussion, it is assumed in this example that the  
14 composite media stream has an audio stream and a video stream.

15 Each media stream has a timeline, and the timelines of the individual  
16 streams are synchronized with each other so that the streams can be rendered in  
17 combination to produce coordinated multimedia content at the network client 11.  
18 The original timelines correspond to the original recording or rendition of the  
19 multimedia material, so that rendering the streams according to their timelines  
20 results in presentation speeds that closely match the speed of the original event or  
21 performance. In the case of audio streams, the timelines preserve the original  
22 speed and pitch of the original audio content.

23 The client computer has a demultiplexer component 104 that receives the  
24 composite media stream and that separates out the individual media streams from  
25



1 the composite format in which the data is streamed (such as ASF). This results in  
2 an audio stream 105 and a video media stream 106. The individual media streams  
3 are sent to and received by respective decoders 108 and 109 that perform in  
4 accordance with the particular data format being employed. For example, the  
5 decoders might perform data decompression.

6 The decoded data streams are then sent to and received by time  
7 modification components: an audio timeline modification component 110 and a  
8 video timeline modification component 111. These components receive input  
9 from a human operator in the form of a speed designation as described above. The  
10 timeline modification components change the timelines of the received media  
11 streams in accordance with the user speed designation to achieve either linear time  
12 compression or linear time expansion. With some types of media, such as video  
13 streams, this involves either omitting selected frames or modifying the  
14 presentation times of the individual data units or video frames. In other cases,  
15 such as with audio streams, some type of audio processing technique as the SOLA  
16 technique described above is used to time-compress or time-expand audio streams,  
17 while maintaining the original pitch of the audio and to also retain the  
18 intelligibility of the audio.

19 The timeline modification components 110 and 111 produce individual  
20 media streams that are provided to and received by respective renderers 114 and  
21 115. The rendering components render the streams in accordance with their  
22 modified timelines, as the streams continue to be streamed from the network  
23 server. In alternative embodiments of the invention, timeline modification  
24  
25



1 components 110 and 111 might be eliminated and their functions performed by  
2 decoders 108 and 109.

3 Note that the speed designation, provided by the user, dictates the rate at  
4 which the network client consumes the composite data stream. Because of this,  
5 the client communicates the speed designation to the network server when  
6 requesting a particular composite media stream. The server responds by streaming  
7 the composite media stream at a rate that depends on or is proportional to the  
8 speed designation provided by the user. For example, for a speed factor of 2.0, the  
9 client consumes data at twice the normal rate. Accordingly, the server streams the  
10 composite media stream at twice its normal rate to meet the demands of the client.

11 In the described embodiment, the user is allowed to change the speed  
12 designation during rendering of the composite media stream. In some cases,  
13 however, it may not be possible to change the playback speed without interrupting  
14 the playback momentarily. If this is the case, playback resumes as soon as  
15 possible, beginning at a point that shortly precedes the point at which playback  
16 was discontinued. Thus, there is some overlap in the presentation—when the  
17 presentation resumes, the overlap provides context for the new content that  
18 follows.

19 Fig. 4 illustrates methodological aspects of the invention. Steps performed  
20 at network client 11 are shown on the left-hand side of the figure, while steps  
21 performed by network server 10 are shown on the right-hand side of the drawing.

22 A step 130 comprises selecting multimedia content from the network server  
23 to be rendered at the network client. In most cases, a user performs this selection  
24 from a menu of available content or via a URL (uniform resource locator)  
25



1 selection. The content will normally be represented by a single composite media  
2 stream. In some cases, different media streams might be available for a particular  
3 content segment, varying perhaps in quality and in required bandwidth.  
4 Preferably, however, the user is unaware of anything except the simple act of  
5 selecting a single topic or composite stream.

6 A step 132 comprises accepting a speed designation from a human user.  
7 This step is independent of the previous step of selecting the content itself.  
8 Furthermore, the user can vary the speed designation at any time during  
9 presentation of the selected content, without having to re-select the content.

10 A step 134 comprises requesting the selected content from the server at a  
11 speed which will satisfy the requirements of the user's speed designation.

12 A step 136, performed by server 10, comprises identifying the particular  
13 composite media stream corresponding to the selected content. Step 138  
14 comprises streaming the composite media stream to the client. In this  
15 embodiment, the composite media stream has its original timeline, which does not  
16 necessarily result in the speed that the user has designated for playback.

17 Step 140, performed by the client, comprises receiving the streaming  
18 content. Step 142 comprises modifying the timeline of the composite media  
19 stream in accordance with the speed designation provided by the user. As  
20 described above, this involves modifying the timelines of the individual media  
21 streams while maintaining their synchronization and intelligibility. Step 144  
22 comprises rendering the composite media stream in accordance with its modified  
23 timeline.



## User Experience

The functionality described above is exposed through an application program executed at network client 11, referred to herein as a streaming multimedia player. The streaming multimedia player may be incorporated into the operating system or run as a separate, self-contained application. In either case, the streaming multimedia player operates in a graphical user interface windowing environment such as provided by the “Windows” brand of operating systems, available from Microsoft Corporation of Redmond, Washington..

Fig. 5 shows one implementation of a graphical user interface window 200 for the multimedia player. This UI window 200 has a command bar 202, a media screen 204, shuttle controls 206, a volume control 208, and content information space 210. Command bar 202 lists familiar UI commands, such as “File”, “View”, and so forth.

Media screen 204 is the region of the UI within which the multimedia content is rendered. For video content, the video is displayed on screen 204. For non-visual content, screen 204 displays static or dynamic images representing the content. For audio content, for example, a dynamically changing frequency wave that represents an audio signal is displayed in media screen 204.

Shuttle controls 206 enable the user to control play of the multimedia content. Shuttle controls 206 include multiple play buttons 212(1), 212(2), and 212(3), a stop button 214, a pause button 216, rewind buttons 218 and 220, and fast forward buttons 222 and 224.

Play buttons 212(1)-212(3) are associated with different playback speeds of the multimedia content. In this illustration, play button 212(1) corresponds to a



1 normal playback speed (i.e., "x1.0"), play button 212(2) corresponds to a faster  
2 playback speed with a speed up factor of 25% (i.e., "x1.25"), and play button  
3 212(3) corresponds to an even faster playback speed with a speed up factor of 50%  
4 (i.e., "x1.50"). It is noted, however, that more or less than three buttons may be  
5 used (e.g., two, four, five, etc.) and may correspond to speeds both above and  
6 below the normalized speed of "x1.0".

7 The user can actuate one of the play buttons via a UI actuation mechanism,  
8 such as a pointer 226 or by tabbing to the desired play button and hitting the  
9 "enter" key. Upon selection of a play button, the multimedia player plays the  
10 multimedia content at the playback speed associated with the selected play button.  
11 For instance, if the user selects play button 212(2) with a 25% speedup factor, the  
12 multimedia player plays the content at a playback speed of 1.25 times the original  
13 or default playback speed.

14 Once the multimedia content is playing at one speed, the user is free to  
15 select a new speed by actuating another of the play buttons 212(1)-212(3).  
16 Suppose the user decides to slow the content back to normal speed. The user can  
17 actuate the "x1.0" play button 212(1) to return the media content to the normal  
18 speed. In response to speed changes, the multimedia player is configured to repeat  
19 a portion of the multimedia content at the new speed.

20 Content information space 210 lists information pertaining to the  
21 multimedia content being rendered on the media screen 204. The content  
22 information space includes the show name, author and copyright information, and  
23 tracking/timing data.  
24  
25



1        Fig. 6 shows another implementation of a graphical user interface window  
2        240 for the multimedia player. Like UI 200 of Fig. 5, UI 240 has command bar  
3        202, media screen 204, shuttle controls 206, volume control 208, and content  
4        information space 210. This implementation, however, employs only a single play  
5        button 242. Actuation of play button 242 initiates play of the multimedia content.

6        UI 240 has a scale mechanism 244 to vary the speed of the content during  
7        rendering. The scale mechanism has a range of playback speeds 246, which in this  
8        example range from 0.5x to 2.5x the normal speed. Scale mechanism 244 also has  
9        a movable slider 248 that is movable over the range 246. The user can position  
10       the slider 248 at the desired speed at which the multimedia player is to play the  
11       multimedia content.

12       In the Fig. 6 illustration, range 246 is a continuous range from a high  
13       playback speed (i.e., 2.5x) to a low playback speed (i.e., 0.5x). Slider 248 moves  
14       continuously over the range. In other implementations, range 246 is a discrete  
15       range of discrete playback speeds (e.g., 0.5x, 1.0x, 1.5x, 2.0x, and 2.5x) and the  
16       slider is movable among the discrete playback speeds.

17       Once the multimedia content is playing at one speed, the user is free to  
18       select a new speed by moving the slider 248 to a new speed. In response to use  
19       manipulation of the scale mechanism, the multimedia player repeats a portion of  
20       the multimedia content and begins playing at the new speed.

21       Fig. 7 shows a third implementation of a graphical user interface window  
22       260 for the multimedia player. In this implementation, UI 260 has a single play  
23       button 262 to initiate playback of the multimedia content. UI 260 also has a menu  
24       264 associated with the play button. In this illustration, menu 264 is a drop-down  
25



1 or pull-down menu that opens beneath the play button in response to actuation of a  
2 tab 266 adjacent to the play button. Alternatively, menu 264 may be invoked by  
3 placing pointer 226 over play button 262 and right clicking a mouse button.

4 Menu 264 lists multiple playback speeds from which a user can select. In  
5 the illustrated example, five playback speeds are listed: x0.5, x0.75, x1.0, x1.25,  
6 and x1.5. The user can select one of the listed speeds to instruct the multimedia  
7 player to play the multimedia content at a desired speed. As noted above, the user  
8 can select a new speed after the content has begun playing by invoking the menu  
9 and selecting the new speed. In response, the multimedia player repeats a portion  
10 of the multimedia content and begins playing at the new speed.

#### 11 12 **Server-Based Multimedia Time-Scale Modification**

13 In various embodiments of the invention, the step of modifying the timeline  
14 of the requested multimedia content can be performed in the client as described  
15 above, in the server, or in both the client and server. In the network environment,  
16 it is desirable to avoid performing any significant timeline modification in the  
17 server. Otherwise, the server could quickly become overloaded with requests from  
18 multiple clients.

19 However, in some cases it may be desirable to store multiple versions of  
20 media streams at a server and to select particular versions of the media streams  
21 depending on the timeline requirements of the client, as designated by the user.  
22 One advantage of this method is that it can require comparatively less  
23 communications bandwidth between the server and client.



1 As a general example, a server might store a plurality of media streams  
2 having timelines modified by different factors. When a client requests a  
3 composite media stream, the server selects the version of the media stream whose  
4 timeline most closely accords with the speed designation set by the user. If the  
5 timeline does not exactly match the speed designation, the client can perform  
6 further timeline modification.

7 Fig. 8 illustrates a more specific example. In this embodiment, a server 300  
8 stores multiple media streams 301 corresponding to specific multimedia content  
9 302. The media streams are of different types, such as audio and video. In Fig. 8,  
10 audio streams are designated by the letter "A" and video streams are designated by  
11 the letter "V". Any combination of a single audio stream and a single video  
12 stream can be rendered to produce the multimedia content.

13 The various individual data streams have timelines that are modified by  
14 different degrees. The speed factors are indicated in Fig. 8. In this embodiment,  
15 the audio and corresponding video streams are organized as pairs, each pair  
16 forming a composite media stream having a timeline that has been modified by a  
17 factor of 0.5, 1.0, or 1.5.

18 When a client 303 requests multimedia content from server 300, the client  
19 identifies both the content and the speed factor. In response, the server selects the  
20 audio and video streams that have timelines most closely approximating the  
21 identified speed factor, and combines those individual media streams to form the  
22 composite media stream. The resulting composite media stream is then sent to the  
23 client. When the timeline is accelerated, this saves bandwidth in comparison to  
24  
25



1 sending an unaltered composite media stream having a higher streaming rate to  
2 meet the accelerated consumption demands of the client.

3 As a further optimization, the server can store composite media streams  
4 having different degrees of timeline modification and different degrees of quality.  
5 Generally, a media stream of a lower quality will consume less communications  
6 bandwidth than a media stream of a higher quality. Before selecting an  
7 appropriate media stream, the server determines the available bandwidth between  
8 the server and the client. It then selects a combination of individual media streams  
9 that provides the best quality while requiring no more than the available  
10 bandwidth.

11 When the user changes the playback speed, the client requests a new media  
12 stream that most closely corresponds to the requested speed. Playback is resumed  
13 in the new stream at the same point (relative to the subject content) at which it was  
14 discontinued in the old stream. Thus, the new stream is initiated at some  
15 intermediate point rather than at the beginning. When the streams are linearly  
16 altered, it is not difficult to determine the appropriate presentation time in the new  
17 stream. Specifically, the point in the new timeline equals  
18  $oldtime(oldfactor/newfactor)$ , where *oldtime* is the presentation time in the first  
19 media stream at which the speed change is to occur, *oldfactor* is the playback  
20 speed or factor of the old media stream, and *newfactor* is the playback speed or  
21 factor of the new media stream.

22 Fig. 9 illustrates a further example utilizing this concept. In this case, a  
23 server 310 has stored a single audio stream 311 and multiple video streams 312, all  
24 corresponding to a single multimedia segment 313. The video streams different in  
25



1 quality and corresponding required bandwidth: low (lo), intermediate (med), and  
2 high (hi). However, the video streams all have a common, unmodified timeline.

3 When a client 314 requests the multimedia content from server 310, the  
4 server determines or notes both the speed factor designated by the user and the  
5 available bandwidth. It then selects the video stream that has best available  
6 quality while also requiring no more bandwidth (at the requested speed factor)  
7 than the difference between the available bandwidth and the bandwidth consumed  
8 by the selected audio stream. Again, this allows the system to compensate for  
9 various available bandwidths.

10 Fig. 10 shows another example, in which a server 320 has stored multiple  
11 audio streams 321 and multiple video streams 322, all corresponding to a single  
12 multimedia segment 323. The audio streams differ in the degree by which their  
13 timelines have been modified. In this example, there are audio streams having  
14 timelines modified by factors of 0.5, 1.0, and 1.5. The video streams differ in  
15 quality and corresponding required bandwidth: low (lo), intermediate (med), and  
16 high (hi). However, the video streams all have a common, unmodified timeline.

17 When a client 324 requests the multimedia content from server 320, the  
18 server determines or notes both the speed factor designated by the user and the  
19 available bandwidth. It then selects an audio stream that most closely accords  
20 with the specified speed factor. It then selects the video stream that has best  
21 available quality while also requiring no more bandwidth than the difference  
22 between the available bandwidth and the bandwidth consumed by the selected  
23 audio stream. Again, this allows the system to compensate for various available  
24 bandwidths.



1        Fig. 11 illustrates yet another embodiment in which multiple media streams  
2 are stored at the server for use depending upon available bandwidth and upon the  
3 speed factor designated by the user. In this embodiment, a server 330 stores a  
4 single audio stream 332 and a plurality of video streams 333, all corresponding to  
5 and representing the same multimedia content 331. The audio stream has an  
6 unaltered timeline. However, the video streams have different timelines and also  
7 vary by quality and corresponding bandwidth requirements. Specifically, in this  
8 example the video streams have timelines modified by factors of 0.5, 1.0, and 2.0.  
9 For each speed factor, there is a “low” bandwidth video stream having a relatively  
10 low quality, and a “high” bandwidth video stream having a relatively high quality.

11        At a normal, unaltered playback rate, the audio stream utilizes a bandwidth  
12 of 16 Kbps (kilobits per second). The low bandwidth video streams require a  
13 bandwidth of 20 Kbps. The high bandwidth streams require a bandwidth of 40  
14 Kbps.

15        Now, suppose that a client requests the multimedia content over a  
16 communications channel having a bandwidth of 56 Kbps, at a speed factor of 2.0.  
17 At this speed factor, the client consumes audio data at twice the normal rate, which  
18 in this case is 32 Kbps. That leaves 24 Kbps of available bandwidth. Accordingly,  
19 the server selects the low bandwidth video stream with the timeline modified by a  
20 factor of 2.0, and combines it with the audio stream to form a composite media  
21 stream for streaming to the client. The total required communications bandwidth  
22 is 52 Kbps, which is within the limits of the available bandwidth.

23        Although the example give with reference to Fig. 11 is relatively specific,  
24 this method of bandwidth utilization can be generalized to include other types of  
25



1 media streams. Each stream is assigned a priority. Audio will generally have a  
2 high priority. The high-priority streams are given priority when allocating  
3 bandwidth. Thus, in the example above, the audio stream is streamed to the client  
4 at its full quality, while the video stream is reduced in quality to fit within the  
5 remaining bandwidth.

6 Furthermore, a stream such as a video stream can sometimes be timeline-  
7 modified dynamically at the server without incurring significant overhead.  
8 Accordingly, the server can adjust the timeline and quality of the video stream  
9 dynamically to match the available bandwidth, eliminating the need to store  
10 multiple video streams at the server. As an example of a situation where this  
11 might be easily accomplished, an MPEG (Motion Picture Expert Group) video  
12 stream contains independent frames and several levels of dependent frames. One  
13 easy way to reduce bandwidth is to simply drop lower-level dependent frames  
14 from the video stream.

## 15 16 **Conclusion**

17 The described methods provide efficient ways to accomplish timeline  
18 modification in a networked client/server environment. The invention provides a  
19 user with a degree of efficiency and convenience that has previously been  
20 unavailable. When using the invention, the user can browse or skim through  
21 streaming content, thereby reducing the time required to assimilate such content.  
22 Furthermore, the invention provides for efficient use of limited bandwidth, thereby  
23 maximizing the practical applications of the invention.



1        Although the invention has been described in language specific to structural  
2 features and/or methodological steps, it is to be understood that the invention  
3 defined in the appended claims is not necessarily limited to the specific features or  
4 steps described. Rather, the specific features and steps are disclosed as preferred  
5 forms of implementing the claimed invention.  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25